

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION ESDS Reuse Working Group

Reuse Tradeoffs: A Case Study of Software Reuse within the NPP Mission

Abstract

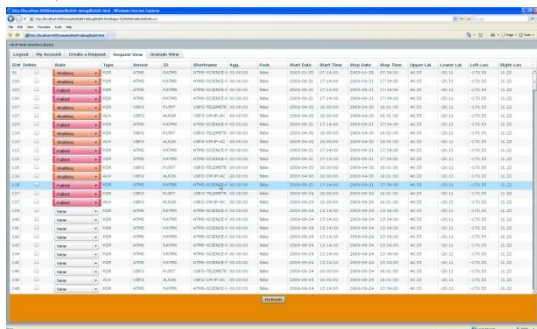
The Science Data Depository and Distribution Element (SD3E) supports the NPOESS Preparatory Project by delivering satellite data products to various climate analysis groups. Within this system, many software components are reused. Conversely, many components were developed using newer programming languages. With *these newer programming languages*, existing functionality must be re-architected in using new paradigms; however, these new frameworks reduced the level of effort in terms of maintenance and adapting to requirements changes. Additionally, some of these new paradigms make these components inherently more reusable. Specifically, this paper will examine the benefits of integrating Adobe Flex and the Spring Framework into the SD3E and as well as the benefits from using reusing code written older programming languages. Additionally, we will examine how our user-interface can be extended to support the Earth Science Community for other missions. The SD3E web interface can support a variety of data formats and may have applications for other Earth Science Missions.

Components NOT Reused

Example	Old Code	New Code
JSP Request Code	The old code had a significant amount of code to generate each column and implementing a sorting feature would have to be done manually.	It is only a single line of code to add a column and sorting and dragging is done automatically. Flex provided many built-in functions that fulfill requirements and could be maintained more easily than reusing old code.
JSP Granule Display Code	Impractical to Implement	Flex offered libraries suited to display data as granules.

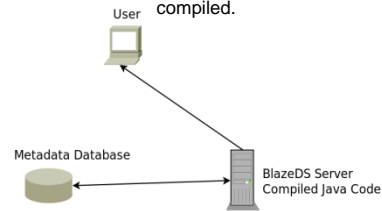
Some code was not reused from one build to another because features could be more easily implemented in other languages, which have better maintainability.

NPP SD3E Request Interface



General NPP SD3E Website Architecture

Flash pages are served to the user via the BlazeDS Server. The Java code uses XML schemas to generate code that later will be compiled.



Components Reused

Example	Description	Reason Reused
Web page design	Includes modules, button layouts, tables, data display, etc.	The original web page design was peer-reviewed by user community and created to meet their needs, so it could be reused as-is.
Workflow	Covers the process by which changes are made and entered in the system.	The user community wanted specific checks at particular points in the process, so this peer-reviewed workflow could also be used as-is.
Back-end Java code	All supporting code for functions present on the web site.	Since these were already developed and tested to work properly, changes in the front-end did not require back-end modifications.

While none of the front-end code was reused, we reused much of the design and feedback received on the original front-end to develop the current version of the web site.

Authors:

Ryan S. Gerard*, Innovim / NASA GSFC
(Ryan.Gerard@nasa.gov)

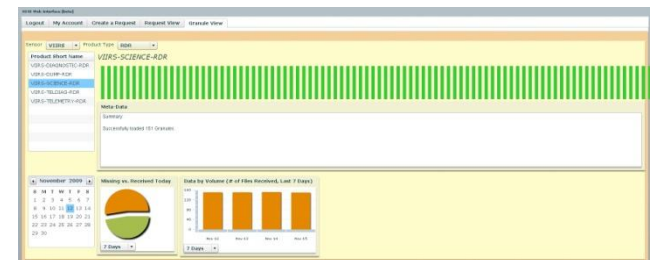
James J. Marshall, Innovim / NASA GSFC

* Presenter

Components Built for Future Reuse

Type of Code	Lines of Code	Possible Reasons for Future Reuse
Adobe Flex Code / Actionscript Code .mxml / .as	2200 (700 reused)	<ul style="list-style-type: none"> The visualizations change automatically to fit the data. All combo boxes are populated dynamically based on XML schemas. No mission-specific data is hardcoded to the interface.
Java Code/XML Schemas/Ant Scripts .java .xml	11000 new code plus 1500 reused for Log4J	Java code is generated based on XML schemas using the Castor open source tool. This code is later compiled. Therefore, when the mission changes requirements, no software updates must be made; updating the XML schemas will automatically update the code.
Database (PostgreSQL Database)	15 tables	Only very generic metadata is stored in our database and much of it could apply to virtually any Earth Science mission.

NPP SD3E Granule Display



Function Name	Description	How Generic is this Component?
Making a Data Request	Request Data for a specific time range, specifying geospatial sub-settings, aggregation, and packaging options, ...	80%
Viewing Current Requests	View Status of Requests (Pending, Failed, Successful), Sort Requests by any Field, ...	95%
Viewing Meta-Data Available	View Files Available for Download, Viewing Meta-Data, Examine Percent of Data Received, ...	95%
User Preferences	Change Password, Manage Email Contact Information, ...	95%

In general, most components are tied very loosely to mission-specific requirements that reuse and/or changes to the display can be implemented easily. This interface could be applied to almost any Earth Science mission that has very basic meta-data available via database. Newer technologies (e.g., Adobe Flex, Actionscript, Java) allow our system to be packaged easily for reuse or tailored for other missions with ease.